The ATLAS event filter

H-P. Beck, C.P. Bee, C. Boissat, M. Caprini, P-Y. Duval, F. Etienne, R. Ferrari, D. Francis, F. Hemmer, R. Jones, et al.

The ATLAS Event Filter

H.P. Beck[1], C.P. Bee[2], M. Caprini[3], P-Y. Duval[2], F. Etienne[2], R. Ferrari[4], D. Francis[3], R. Jones[3], D. Laugier[2], L. Mapelli[3], C. Meessen[2], R. Mommsen[1], G. Mornacchi[3], R. Nacasch[2], Z. Qian[2], C. Rondot[2], F. Touchard[2], V. Vercesi [4]

[1] Laboratory for High Energy Physics, University of Berne, Switzerland,
[2]Centre de Physique des Particules de Marseille, France,
[3]CERN, Geneva, Switzerland,
[4] DNTP, University of Pavia and INFN, Italy

Presented by C.P. Bee

Abstract

The Event Filter is the last element of the ATLAS trigger system. It will comprise a farm of software processors running full offline algorithms, with access to the fully built raw event data, to reduce the event rate by a factor of ten before writing the data to mass storage. The strategies currently adopted to develop software and hardware prototypes based on PCs and SMP machines are discussed as well as some of the issues involved in constructing a computing engine of sufficient size to fulfil the ATLAS requirements.

1 Introduction

The ATLAS Event Filter (EF) implements the final (third) level of event processing and rejection before mass storage. Unlike the Level 1 [1] and Level 2 [2] triggers, which employ specialised algorithms with total latencies of 2μs and 10ms respectively on highly selective event data, the EF [2] will use offline physics and event reconstruction algorithms accessing the full event data and is expected to have a latency of a few seconds. The EF is situated directly in the data acquisition chain after the Event Builder (EB) (fig. 1). It will comprise a set of event processing “sub-farms”, each connected to an output port of the EB switch. The data processing and local control will be independent between sub-farms, but they will be controlled by a single global EF supervisor which itself will be under control of the experiment’s online run control system. The possible physical implementations of the hardware which will be used to construct the EF as well as the high level design of the control, monitoring and physics application software is presently under study. Various hardware candidates will be considered by constructing prototypes and comparing their relative performance according to criteria which are currently being developed. General architecture types which will be considered are: farms based on commodity components (PCs, Ethernet switches) and SMP-type machines.

Figure 1 ATLAS Global DAQ Architecture

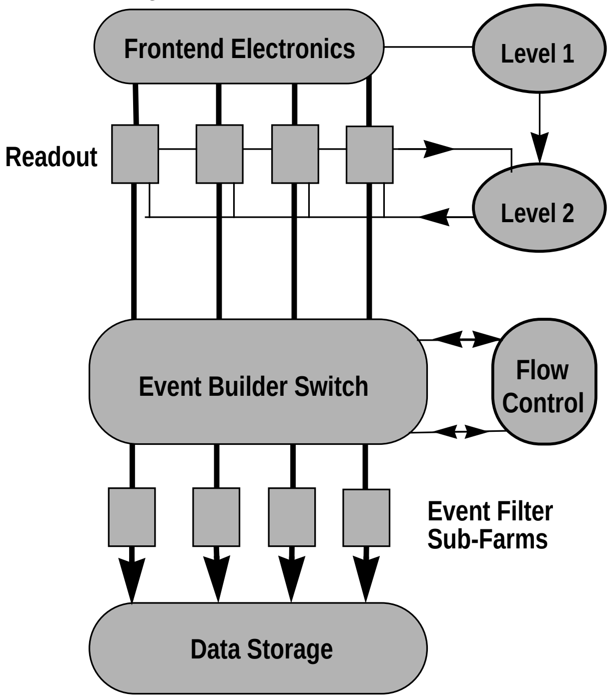
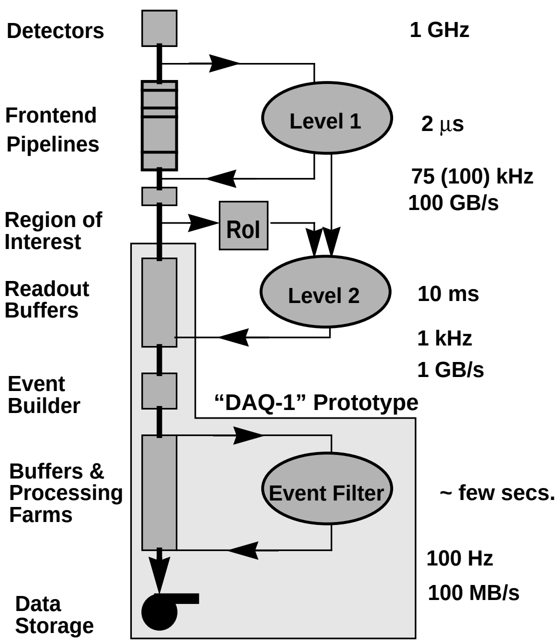


Figure 2 ATLAS Global Trigger Architecture



2 Global Requirements and Constraints

2.1 ATLAS Trigger/DAQ Global Architecture

Schematic views of the ATLAS Trigger and DAQ architecture are shown in figs. 1 and 2. The initial interaction rate of 1 GHz (bunch crossing rate of 40 MHz with ~ 25 interactions per bunch crossing) is reduced to ~ 40 kHz by the LVL1 trigger making an initial selection based on reduced-granularity information from a subset of detectors. The ATLAS front-end systems are designed to accept a maximum LVL1 rate of 75 kHz, which could be upgraded to 100 kHz.

Events chosen by the LVL1 are read out into the readout buffers (ROBs) from where they are accessed by the LVL2 trigger. The LVL2 makes use of the so-called ‘region of interest’ (RoI) information from the LVL1 in order to allow selective access of event data in the ROBs. This RoI data amounts to only a few percent of the full event data. Events accepted by the LVL2 (at an expected rate of ~ 1 kHz) are then built by the Event Builder and transferred for subsequent treatment to the Event Filter. The EF makes a final event selection and reduces the rate of events written to mass storage to ~ 100 Hz, with an expected total event size of ~ 1.3 MB.

2.2 Event Filter Requirements

The EF is different from the lower level triggers in several important respects, namely; it has access to the complete event data, it is planned to use the reconstruction and physics algorithms as directly as possible from the offline (as opposed to developing specialised algorithms), and it will have a latency of at least a factor of 100 larger than that of the LVL2. The reduction of the data to be written to permanent storage will be performed by rejecting events which do not conform to the physics requirements, by minimising the size of the accepted events (it may well be that e.g. some classes of accepted events will not require to have all the raw data output to storage) and by employing some data compression techniques.

Given that it has access to complete event data, the EF is also in a position to perform global monitoring, calibration and alignment functions online, which are not possible at the detector readout level. This will be a vital element in the overall quality control of the experiment both for the physics quality and the detector optimisation and performance. The quality of the detector calibration will of course have a direct bearing on the quality of the EF decision itself.

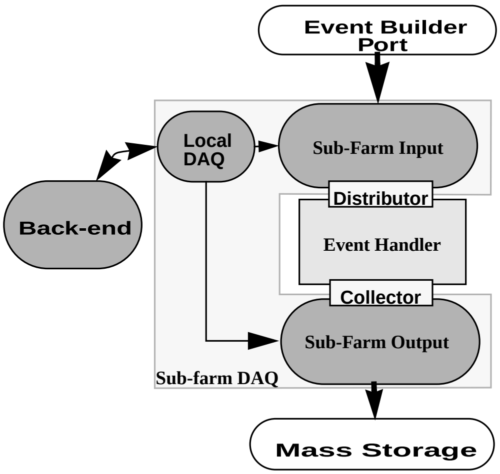
Based on extrapolations from other experiments and from current ATLAS physics studies, the EF is expected to require a total computing power in excess of 25 kSPECint95, which translates to an equivalent of 1,000 CPUs with a capacity of 1,000 MIPS each. This is more than a factor of ten larger than the computing power of level 3 trigger processors in the current generation of experiments (e.g. CDF) [3].

3 Event Filter Architecture

3.1 Global Architecture

The global architecture of the EF is shown in fig. 3. It will comprise a number of sub-farms, each connected independently to the EB [4]. Each sub-farm in turn will comprise a number of processing nodes, interface elements (known as the Distributor and Collector) between these nodes and the data acquisition dataflow, and a sub-farm supervisor. Some of the Distributor and Collector functionality will be implemented on the dataflow elements: SFI and SFO, the remainder, together with the processing elements and the Supervisor are collectively known as the Event Handler (EH). This division has been done in order to make the EH an independent object which can be designed, developed and tested without the need for support from the rest of the DAQ system. The dataflow elements are interfaced to the EH by an API [5] implemented by the Distributor and Collector. Fig 3 illustrates this layout. The sub-farm DAQ [6] (fig. 3) is responsible for supplying data to and receiving data from the EH, and is controlled by the Local DAQ [7], through which the interface to the online system is also implemented (Backend, fig. 3) [8]. It is also responsible for receiving the data from the EB and, after treatment by the EH, passing accepted events to the mass storage system.

Figure 3 EF Global Architecture



3.2 Control Software - High Level Design

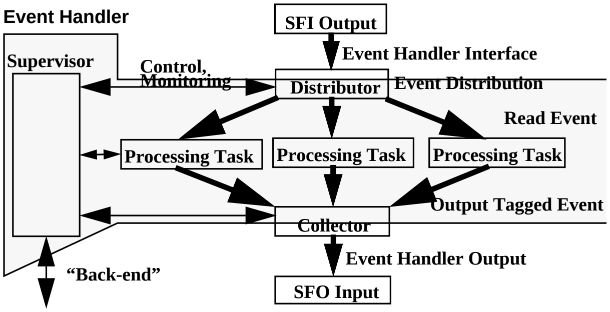
A high level design for the EH control software has been made [9], taking into account the following key requirements:

- Independence from hardware architecture choice
- Data security during passage through the EH (recoverability)
- Event distribution within the EH according to trigger characteristics
- Data driven

Different implementations (e.g. on different platforms) based on the same global design will then enable direct comparisons to be made.

A block diagram of the high level design is shown in fig. 4. The Distributor, on reception of an event from the SFI, sends it immediately to a “safe storage” location (e.g. disk file) and classifies it according to its event “type” (defined by the trigger characteristics of the event). According to this type, the event is then sent to the input queue of a group of “Processing Tasks” (PT) which have been initialized to deal with such an event type. When free, one of these PTs accesses the event and processes it. The PTs implement the offline reconstruction and physics algorithms which effects the event filtering. If the event is accepted, it will be passed on to the Collector which gathers together accepted events from all PTs in the EH and stores them on a further area of safe storage to await transfer to the SFO and subsequently the mass storage system. Once the event has been safely dispatched to the SFO, backup copies of it in the safe storage areas are deleted. If the event fails the physics cuts imposed by the algorithms, it will be rejected, and the backup “copy” will be removed from the safe storage. In the event of a failure during event transfer or treatment in the EH, an event may always be recovered from the safe storage and re-treated, or passed to a special data stream for in depth analysis of the possible causes of the failure. The design is entirely data driven, the Supervisor component intervening only for issues such as process control, initialisation, status information access, error handling and reporting, online run parameter access and overall EH control. The Supervisor is also the interface between the EH and the global online system.

Figure 4 Block Diagram of Control Software



The design is generic and will be implemented on various different hardware prototypes in order to compare the performance. A description of the implementation of this design on a PC based prototype is given elsewhere in these proceedings [10].

3.3 Processing Task Strategy

The EF will use event reconstruction and physics analysis algorithms which have been developed for the offline software of ATLAS. It will receive events which have been accepted by the LVL1 and LVL2 triggers. Its first task therefore will be to confirm and refine these decisions based on its access to the entire event data and improved calibration constants as well as the more complex algorithms which it will have at its disposal. Having confirmed a LVL2 decision, the EF will then proceed to apply more global physics selection algorithms in order to reduce the final output event rate. A summary of the physics requirements for the EF can be found in [2].

4 Prototypes and Development Strategy

4.1 Prototypes

In order to prepare the final EF design and architecture choices which are due in 2001, a large amount of prototype work is planned, to be able to weigh up the relative merits of the various possible hardware and software solutions. The two major hardware options under prototype study are: farms based on commercially available PCs [11], and proprietary solutions based on symmetric multi-processor (SMP) machines [12]. Issues including: scalability from the prototype size to that of the final system (with the help of dynamic system modelling), farm management and control, error handling, reliability and robustness all need to be studied and evaluated on different platforms.

4.2 Development Strategy

As well as evaluating the various hardware options, the prototypes under construction will serve as bases for testing the control software design described above. Each prototype will be integrated as a standalone EF sub-farm into overall DAQ-1 prototype [2] system which is being developed for ATLAS, and which will be installed in a testbeam environment in 1999. This will enable us to make tests of the overall operation and control of the prototypes within the DAQ-1 context in quasi-realistic conditions. In parallel to the development of the software and architectural issues, benchmark programs based on real ATLAS event reconstruction and physics analysis software are being prepared and used in the prototype farm processing tasks described above. Studying these benchmarks on the prototypes will enable us to gain a better view of the resource utilisation and therefore of the final configuration of the global EF in terms of memory, disk, network and cpu capacity. Results from this prototyping phase will provide vital input to the final design decisions which have to be made in 2001.

5 Conclusions

The general architecture of the ATLAS trigger & DAQ system has been presented with emphasis on the role of the Event Filter third level trigger system and its design requirements. The global control software architecture and design was described and the project development plans for the forthcoming 18 months were outlined, with the construction of prototypes to study hardware and software architecture issues playing a key role in this development.

6 References

[1] First Level Trigger Technical Design Report, CERN/LHCC/98-14, 30 June 1998

[2] ATLAS DAQ, EF, LVL2 and DCS Technical Progress Report, CERN/LHCC/98-16, 30 June 1998

[3] J. Fromm *et al.*, “Processing Farms Plans for CDF and D0 Run II”, these proceedings

[4] R. Spiwoks *et al.*, “Event Building in the ATLAS Prototype DAQ”, these proceedings

[5] The Event Handler API, ATLAS DAQ-1 Note 98 http://atddoc.cern.ch/Atlas/Notes/098/Note098-1.html

[6] The Sub-farm DAQ for the ATLAS DAQ Prototype -1, ATLAS DAQ-1 Note 44 http://atddoc.cern.ch/Atlas/Notes/044/Note044-1.html

[7] The LDAQ in ATLAS DAQ Prototype -1, ATLAS DAQ-1 Note 40 http://atddoc.cern.ch/Atlas/Notes/040/Note040-1.html

[8] ATLAS Back-end Software, ATLAS DAQ Note 87, ATL-DAQ-98-087

[9] A High Level Design for the Sub Farm Event Handler, ATLAS DAQ-1 Note 61 http://atddoc.cern.ch/Atlas/Notes/061/Note061-1.html

[10] C.P. Bee *et al.*, “A Prototype Study for the ATLAS Event Filter Based on a PC Farm”, these proceedings

[11] Proposal for an Event Filter Prototype Based on PC Computers in the DAQ-1 Context, ATLAS DAQ-1 Note 45, http://atddoc.cern.ch/Atlas/Notes/045/Note045-1.html

[12] Proposal for an Event Filter Prototype Based on a Symmetric Multi-Processor Architecture, ATLAS DAQ-1 Note 82, http://atddoc.cern.ch/Atlas/Notes/082/Note082-1.html